On barrier and modified barrier multigrid methods for 3d topology optimization

Brune, Alexander; Kocvara, Michal

DOI:
10.1137/19M1254490

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Brune, A & Kocvara, M 2020, 'On barrier and modified barrier multigrid methods for 3d topology optimization', *SIAM Journal on Scientific Computing*, vol. 42, no. 1, pp. A28–A53. https://doi.org/10.1137/19M1254490

SIAM J. SCI. COMPUT.
Vol. 42, No. 1, pp. A28–A53

ON BARRIER AND MODIFIED BARRIER MULTIGRID METHODS FOR THREE-DIMENSIONAL TOPOLOGY OPTIMIZATION*

ALEXANDER BRUNE† AND MICHAL KOČVARA‡

Abstract. One of the challenges encountered in optimization of mechanical structures, in particular in what is known as topology optimization, is the size of the problems, which can easily involve millions of variables. A basic example is the minimum compliance formulation of the variable thickness sheet (VTS) problem, which is equivalent to a convex problem. We propose to solve the VTS problem by the penalty-barrier multiplier (PBM) method, introduced by R. Polyak and later studied by Ben-Tal and Zibulevsky and others. The most computationally expensive part of the algorithm is the solution of linear systems arising from the Newton method used to minimize a generalized augmented Lagrangian. We use a special structure of the Hessian of this Lagrangian to reduce the size of the linear system and to convert it to a form suitable for a standard multigrid method. This converted system is solved approximately by a multigrid preconditioned minimal residual method. The proposed PBM algorithm is compared with the optimality criteria method and an interior point method, both using a similar iterative solver setup. We apply all three methods to different loading scenarios. In our experiments, the PBM method clearly outperforms the other methods in terms of computation time required to achieve a certain degree of accuracy.

Key words. topology optimization, multigrid methods, interior point methods, augmented Lagrangian methods, preconditioners for iterative methods, modified barrier functions

AMS subject classifications. 65N55, 35Q93, 90C51, 65F08

DOI. 10.1137/19M1254490

1. Introduction.

The goal of topology optimization is to find an optimal geometry of a solid body that maximizes its performance under certain boundary conditions, by determining an optimal distribution of material in a predefined design domain. It has many applications in industry, such as in mechanical and electrical engineering. The main challenge is the high computational cost of solving large-scale systems that arise from numerical methods to solve PDEs on high-resolution meshes. A basic example of topology optimization is the minimum compliance problem, where the deformation energy of an elastic body under prescribed loading and boundary conditions is to be minimized, given an amount of material. Relating the local stiffness of the body linearly to the continuous material distribution and employing a finite element discretization leads to the so-called *variable thickness sheet* (VTS) problem

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 5, 2019; accepted for publication (in revised form) October 4, 2019; published electronically January 7, 2020.

https://doi.org/10.1137/19M1254490

Funding: This work was partially supported by Fondation mathematique Jaques Hadamard FMJH/PGMO project 2017-0088 "Multi-level Methods in Constrained Optimization."

†School of Mathematics, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK (ACB795@student.bham.ac.uk).

‡School of Mathematics, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK, and Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 18208 Praha 8, Czech Republic (m.kocvara@bham.ac.uk).

$$
\begin{aligned}
&\min_{\rho\in\mathbb{R}^m,\, u\in\mathbb{R}^n} \frac{1}{2} f^\top u \\
&\text{subject to} \\
&\qquad K(\rho)u = f, \\
&\qquad \sum_{i=1}^{m} \rho_i = V, \\
&\qquad \rho_i \geq \underline{\rho}_i, \quad i = 1,\dots,m, \\
&\qquad \rho_i \leq \bar{\rho}_i, \quad i = 1,\dots,m\,,
\end{aligned}
\tag{1.1}
$$

where $K(\rho) = \sum_{i=1}^m \rho_i K_i$, with $K_i \in \mathbb{R}^{n\times n}$, is the stiffness matrix and $f \in \mathbb{R}^n$ is the load vector of the finite element equilibrium equations. The design variable ρ is commonly referred to as the *density*, while the vector u represents the nodal displacements. We assume that K_i are symmetric and positive semidefinite and that $\sum_{i=1}^m K_i$ is sparse and positive definite. We also assume that the volume $V \in \mathbb{R}$ and the lower and upper bounds $\underline{\rho} \in \mathbb{R}^m_+$ and $\bar{\rho} \in \mathbb{R}^m_+$ are chosen such that the problem is strictly feasible. This implies $\bar{\rho} > \underline{\rho}$, among other things. While problem (1.1) is not itself convex, it is equivalent to a convex problem; see [6] and Theorem 2.1 below. For a more detailed derivation of the VTS problem and a comprehensive treatment of the theory and applications of topology optimization, see, for example, [8].

The minimum compliance problem has been studied extensively. Still, it is the subject of ongoing research as higher design detail calls for higher mesh resolution, which in turn makes the problem more computationally demanding. Aage et al., for example, performed topology optimization on a model with more than one billion elements [1]. The bottleneck of algorithms for topology optimization is usually the solution of large linear systems. Direct solvers are not a viable option, due to their computational complexity and demand on computer memory, and iterative, most typically Krylov type solvers, are given preference. Since their convergence behavior highly depends on the condition number of the system matrix, preconditioning plays a vital role. The multigrid method, introduced by Brandt as a solver for boundary-value problems [9], has become popular as a means to precondition the system by employing it inside the iterative solvers. As early as 2000, Maar and Schulz [20] proposed a conjugate gradient method preconditioned by multigrid for topology optimization. Similar solvers were used in [2] and [15]. In [1], the authors chose a multilayered algorithm involving two types of Krylov solvers and the geometric as well as algebraic multigrid method. We refer the reader to [10] for a comprehensive introduction to the multigrid method.

Beyond the issue of efficiently solving the linear systems arising within each iteration of the optimization algorithm, the total number of such iterations required to reach the optimal solution—and thus the choice of optimization method—also affects the overall time-efficiency of the algorithm. The most commonly used methods for the minimum compliance problem are the *optimality criteria* (OC) method (see [8]) and the *method of moving asymptotes* (MMA) [25]. One of the advantages of the OC method is that it is relatively simple to implement; see in particular [3]. To our knowledge, global convergence results exist only for the MMA, and it is often the algorithm of choice in commercial software or large-scale applications, such as that described in [1]. Both of these methods, however, usually rely on heuristics for their stopping criteria and, in practice, display a very similar rate of convergence.

A possible alternative to the aforementioned methods is the *interior point* (IP) method. It has become increasingly popular in the past twenty to thirty years, par-

ticularly for convex optimization [27]. Its theoretical advantage over the OC method or the MMA for convex problems lies in its rate of convergence, especially for convex quadratic problems such as (1.1). Maar and Schulz [20] used an IP algorithm for two-dimensional topology optimization. In [14], Jarre, Kočvara, and Zowe proposed an IP method for truss topology optimization. This was later extended in [15] to large two-dimensional VTS problems, where it outperformed the OC method, in terms of both iterations and overall CPU time required to achieve optimality to within a certain precision. In one part of our paper, we build on this work and further improve the algorithm to apply it to large-scale three-dimensional (3D) problems. The approach is described in section 4, and results of some examples are presented in section 7.

Going from two to three dimensions is by no means straightforward. The largest examples in [15] were based on nine regular refinements of a very coarse, e.g., 2×2, mesh. This resulted in 262 144 finite elements and 526 338 degrees of freedom (components of the displacement vector u). Such a problem could still be solved on a standard laptop. If we used the same refinement level in a 3D example starting with a $2 \times 2 \times 2$ coarse mesh, we would end up with a problem with more than 134 million finite elements and 405 million degrees of freedom. Moreover, while the stiffness matrix in two dimensions typically has 18 nonzero elements per row, in 3D problems this number typically goes up to 81 nonzeros, i.e., the stiffness matrix is considerably denser. All this makes much greater demands on the numerical linear algebra used in the optimization algorithm.

A common problem with IP methods is the ill-conditioning of the system as the iterates approach the optimal solution. This leads to an increase in solver iterations which can make the algorithm nonviable. A class of methods that aims to counteract this problem, while otherwise following a strategy similar to that of the IP method, is the class of *penalty-barrier multiplier* (PBM) methods. They were first introduced in [7], building on the modified barrier methods proposed by Polyak in [22]. As part of the larger class of augmented Lagrangian methods, they have one particular convergence property which sets them apart from IP methods. The latter involve a sequence of barrier parameters which needs to tend to 0 for convergence to the optimal solution, this being the cause of the increasing ill-conditioning; the former feature a penalty parameter for which there exists a value larger than 0 such that the method still converges to the optimal solution. See, for example, [24, Corollary 6.15] for a result specific to penalty-barrier methods. PBM methods have been successfully applied to convex problems and semidefinite problems in topology optimization [17]. In section 3 of this paper, a penalty-barrier method for (1.1) is introduced. In contrast to the IP method, the PBM method does not stay in the strict interior of the feasible region. This poses a problem with regard to the positive definiteness of $K(\rho)$, which depends on ρ_i being strictly positive for all $i = 1, \dots, m$. We circumvent this problem by applying the PBM method to the dual of (1.1). The theoretical background for this is covered in section 2. The PBM approach described in section 3 is applied to several examples in section 7, in order to compare it to the IP method from section 4, as well as to the OC method, which is briefly described in section 5.

Lastly, a remark on notation: throughout this paper, we use e_i to denote the ith canonical unit vector and e to denote the vector $(1, \dots, 1)^\top$ of appropriate dimension.

2. Dual VTS problem. Consider the VTS problem (1.1). Following [5, 18] in the context of equivalent formulations for truss topology optimization, we can formulate a dual problem to (1.1):

$$\min_{u\in\mathbb{R}^n,\,\alpha\in\mathbb{R},\,\underline{\nu},\,\bar{\nu}\in\mathbb{R}^m} \alpha V - f^\top u - \underline{\rho}^\top \underline{\nu} + \bar{\rho}^\top \bar{\nu}$$

subject to

$$\begin{aligned} &\frac{1}{2}u^\top K_i u \le \alpha - \underline{\nu}_i + \bar{\nu}_i, \quad i = 1,\dots,m, \\ &\underline{\nu}_i \ge 0, \quad i = 1,\dots,m, \\ &\bar{\nu}_i \ge 0, \quad i = 1,\dots,m\,. \end{aligned} \tag{2.1}$$

THEOREM 2.1. *Problems* (1.1) *and* (2.1) *are equivalent in the following sense:*

(i) *If one problem has a solution, then also the other problem has a solution and*

$$\min(1.1) = \min(2.1)\,.$$

(ii) *Let* $(u^*, \alpha^*, \underline{\nu}^*, \bar{\nu}^*)$ *be a solution to* (2.1)*. Further, let* τ^* *be the vector of Lagrangian multipliers for the inequality constraints associated with this solution. Then* (τ^*, u^*) *is a solution of* (1.1)*. Moreover,*

$$\underline{\nu}_i^* \bar{\nu}_i^* = 0, \quad i = 1,\dots,m.$$

(iii) *Let* (ρ^*, u^*) *be a solution of* (1.1)*. Further, let* $\underline{r}^*$ *and* $\bar{r}^*$ *be the Lagrangian multipliers associated with the lower and upper bounds on* ρ*, respectively, and let* λ^* *be the multiplier for the volume constraint. Then* $(u^*, \lambda^*, \underline{r}^*, \bar{r}^*)$ *is a solution of* (2.1)*.*

Proof. We will first write (1.1) equivalently as

$$\min_{\substack{\rho_i \le \rho_i \le \bar{\rho}_i \\ \sum_{i=1}^m \rho_i = V}} \max_{u\in\mathbb{R}^n} \; f^\top u - \frac{1}{2}u^\top K(\rho) u\,. \tag{2.2}$$

Indeed, as $K(\rho)$ is by assumption positive semidefinite, the necessary and sufficient optimality condition for the inner maximization problem is $K(\rho)u = f$, and, using this, the optimal value of the maximization problem is $\frac{1}{2}f^\top u$. Problem (2.2) is convex (actually linear) and bounded in ρ and concave in u, so we can switch "max" and "min" (see, e.g., [11]) to get an equivalent problem:

$$\max_{u\in\mathbb{R}^n} \inf_{\substack{\rho_i \le \rho_i \le \bar{\rho}_i \\ \sum_{i=1}^m \rho_i = V}} f^\top u - \frac{1}{2}u^\top K(\rho) u\,.$$

Due to our assumption of strict feasibility, there exists a Slater point for the feasible set of the inner (convex) optimization problem, so we may replace it by its Lagrangian dual. The Lagrangian multipliers for the inequalities will be denoted by $\underline{r} \in \mathbb{R}^m_+$ and $\bar{r} \in \mathbb{R}^m_+$, that for the volume equality constraint by $\lambda \in \mathbb{R}$:

$$\max_{u\in\mathbb{R}^n} \max_{\substack{\lambda\in\mathbb{R} \\ \underline{r}\in\mathbb{R}^m_+, \bar{r}\in\mathbb{R}^m_+}} \inf_{\rho\in\mathbb{R}^m_+} \; f^\top u - \frac{1}{2}u^\top K(\rho) u + \lambda\Big(\sum_{i=1}^m \rho_i - V\Big) - \underline{r}^\top(\rho - \underline{\rho}) + \bar{r}^\top(\rho - \bar{\rho})\,. \tag{2.3}$$

We can include the nonnegativity constraint on ρ in the innermost optimization problem because we know that the solution to (2.3) satisifies $\rho \ge \underline{\rho} \ge 0$.

Now regard the dual problem (2.1). It can equivalently be formulated as the following min-max problem, using a partial Lagrangian function with multiplier $\tau \in \mathbb{R}^m$:

$$\min_{\substack{u\in\mathbb{R}^n \\ \alpha\in\mathbb{R} \\ \underline{\nu}\in\mathbb{R}^m_+, \bar{\nu}\in\mathbb{R}^m_+}} \max_{\tau\in\mathbb{R}^m_+} \; \alpha V - f^\top u - \underline{\nu}^\top \underline{\rho} + \bar{\nu}^\top \bar{\rho} + \sum_{i=1}^m \tau_i \left(\frac{1}{2}u^\top K_i u - \alpha + \underline{\nu}_i - \bar{\nu}_i\right)$$

which can be rearranged further to give

$$\min_{\substack{u\in\mathbb{R}^n \\ \alpha\in\mathbb{R} \\ \underline{\nu}\in\mathbb{R}^m_+,\bar{\nu}\in\mathbb{R}^m_+}} \max_{\tau\in\mathbb{R}^m_+} \frac{1}{2}u^\top K(\tau)u - f^\top u + \alpha(V - \sum_{i=1}^m \tau_i) + \underline{\nu}^\top(\tau-\underline{\rho}) - \bar{\nu}^\top(\tau-\bar{\rho})\,. \tag{2.4}$$

Identifying τ, α, $\underline{\nu}$, and $\bar{\nu}$ with ρ, λ, $\underline{r}$, and $\bar{r}$, respectively, and changing the sign of the objective function (and thus changing "max" to "min" and "min" to "max"), we can see that (2.3) and (2.4) are equivalent. For later reference, note that the multiplier τ of the dual problem corresponds to the primal variable ρ, the density.

The second part of (ii) is obvious from the fact that $\underline{\nu}$ and $\bar{\nu}$ are multipliers for the lower and upper bounds, so only one of them can be positive (only one bound can be active) for each component. ∎

Notice that (2.1) is a convex optimization problem, as K_i are positive semidefinite.

We finish this section with another formulation of the dual VTS problem that allows us to easily compute the duality gap (this formulation was first derived in [5]).

THEOREM 2.2. *Problem* (2.1) *is equivalent to an unconstrained nonsmooth problem*

$$\max_{u\in\mathbb{R}^n,\alpha\in\mathbb{R}} -\alpha V + f^\top u + \sum_{i=1}^m \min\{(\alpha - \frac{1}{2}u^\top K_i u)\underline{\rho}_i, (\alpha - \frac{1}{2}u^\top K_i u)\bar{\rho}_i\} \tag{2.5}$$

in the following sense:

(i) $\min(2.1) = -\max(2.5)$*;*

(ii) *Let* $(u^*,\alpha^*,\underline{\nu}^*,\bar{\nu}^*)$ *be a solution of* (2.1)*. Then* (u^*,α^*) *is a solution of* (2.5)*. Conversely, every solution* (u^*,α^*) *of* (2.5) *is a part of a solution of* (2.1)*.*

Proof. We will show that (2.1) and (2.5) are equivalent reformulations of each other. Introducing an auxiliary variable $s \in \mathbb{R}^m$, problem (2.5) can be directly rewritten as

$$\max_{u\in\mathbb{R}^n,\alpha\in\mathbb{R},s\in\mathbb{R}^m} -\alpha V + f^\top u + \sum_{i=1}^m s_i$$

subject to

$$\left(\alpha - \frac{1}{2}u^\top K_i u\right)\bar{\rho}_i \ge s_i, \quad i = 1,\dots,m,$$
$$\left(\alpha - \frac{1}{2}u^\top K_i u\right)\underline{\rho}_i \ge s_i, \quad i = 1,\dots,m\,.$$

The constraints in the above problem can be written as

$$\left(\alpha - \frac{1}{2}u^\top K_i u\right) \ge \max\left\{\frac{s_i}{\underline{\rho}_i}, \frac{s_i}{\bar{\rho}_i}\right\}, \quad i = 1,\dots,m\,.$$

Noting that $\bar{\rho} > \underline{\rho} \ge 0$, we define

$$\underline{\nu}_i = \frac{s_i}{\underline{\rho}_i}\,,\ \bar{\nu}_i = 0 \qquad \text{if} \quad \frac{s_i}{\underline{\rho}_i} > \frac{s_i}{\bar{\rho}_i} > 0,$$
$$\underline{\nu}_i = 0\,,\ \bar{\nu}_i = -\frac{s_i}{\bar{\rho}_i} \quad \text{if} \quad \frac{s_i}{\underline{\rho}_i} \le \frac{s_i}{\bar{\rho}_i} \le 0\,.$$

Then the above set of constraints can also be written as

$$\left(\alpha - \frac{1}{2}u^T K_i u\right) \geq \underline{\nu}_i - \bar{\nu}_i \quad i = 1, \ldots, m\,.$$

Obviously, these $\underline{\nu}_i, \bar{\nu}_i$ also satisfy the nonnegativity constraints. Lastly, we can reformulate the objective function to match (2.1), since

$$\sum_{i=1}^{m} \underline{\rho}_i \underline{\nu}_i - \sum_{i=1}^{m} \bar{\rho}_i \bar{\nu}_i = \sum_{i:\frac{s_i}{\underline{\rho}_i} > \frac{s_i}{\bar{\rho}_i}} \underline{\rho}_i \frac{s_i}{\underline{\rho}_i} + \sum_{i:\frac{s_i}{\underline{\rho}_i} \leq \frac{s_i}{\bar{\rho}_i}} \bar{\rho}_i \frac{s_i}{\bar{\rho}_i} = \sum_{i=1}^{m} s_i\,.$$

We switch the sign of the objective function, and claims (i) and (ii) follow. ∎

Assume that (u, α) is a feasible point in the dual problem (2.1) such that there exist ρ satisfying $K(\rho)u = f$ and (ρ, u) is feasible in the primal problem (1.1). We then have the following formula for the duality gap:

$$\begin{aligned} \delta(u,\alpha) &:= \min(1.1) - \max(2.5) \\ &= -\frac{1}{2}f^\top u + \alpha V - \sum_{i=1}^{m} \min\left\{\underline{\rho}_i\left(\alpha - \frac{1}{2}u^\top K_i u\right), \bar{\rho}_i\left(\alpha - \frac{1}{2}u^\top K_i u\right)\right\}. \end{aligned} \tag{2.6}$$

3. The PBM method for topology optimization. In this section, we describe the class of PBM algorithms and their application to the VTS problem. This class of algorithms was formally introduced by Ben-Tal and Zibulevsky in [7]. At the same time, Polyak and Teboulle discussed a very similar class of nonlinear multiplier methods in [23], using the approach of the nonlinear rescaling (NR) principle and proximal-like mapping, which was based on earlier work by Teboulle [26]. Both classes, PBM and NR, generalize several types of multiplier methods, such as the modified-barrier function method [22]. The PBM class, however, includes a specific class of nonlinear multiplier methods that proved to be computationally very efficient; see also [16]. We use the same type of method in this paper and therefore adopt the nomenclature PBM method. This method was first applied to topology optimization problems in [19].

3.1. PBM methods. Consider a generic convex constraint optimization problem

$$\min_x \{\mathfrak{f}(x) \mid \mathfrak{g}_i(x) \leq 0,\ i = 1, \ldots, m\}\,.$$

The idea of NR and PBM is to replace the inequalities by scaled inequalities $p_i\varphi(\frac{\mathfrak{g}_i(x)}{p_i}) \leq 0$ with a penalty function φ and a penalty parameter $p_i > 0$. Here, φ is a strictly increasing, twice differentiable, real-valued, strictly convex function with dom $\varphi = (-\infty, b)$, $0 < b \leq \infty$, which has the following properties:

- $(\varphi 1)$ $\varphi(0) = 0$,
- $(\varphi 2)$ $\varphi'(0) = 1$,
- $(\varphi 3)$ $\lim_{s\to b} \varphi'(s) = \infty$,
- $(\varphi 4)$ $\lim_{s\to -\infty} \varphi'(s) = 0$,
- $(\varphi 5)$ $\varphi''(s) \geq \frac{1}{M}$ for all $s \in [0, b]$, for some $M > 0$.

Then the "penalized" problem

$$\min_x \{\mathfrak{f}(x) \mid p_i\varphi\left(\frac{\mathfrak{g}_i(x)}{p_i}\right) \leq 0,\ i = 1, \ldots, m\} \tag{3.1}$$

remains convex and has the same feasible set and thus the same solution as the original one. We formulate a standard Lagrangian function of the penalized problem that can be considered an augmented Lagrangian function of the original problem:

$$\mathcal{L}(x,\mu;p) = \mathfrak{f}(x) + \sum_{i=1}^{m} \mu_i p_i \varphi\left(\frac{\mathfrak{g}_i(x)}{p_i}\right). \tag{3.2}$$

At each iteration of the PBM, we minimize the augmented Lagrangian with respect to x,

$$\textit{Step } 1. \qquad x^{k+1} \approx \arg\min_x \mathcal{L}(x,\mu^k;p^k), \tag{3.3}$$

and update the multipliers and the penalty parameter:

$$\textit{Step } 2. \qquad \mu_i^{k+1} = \mu_i^k \varphi'\left(\frac{\mathfrak{g}_i(x^{k+1})}{p_i^k}\right), \tag{3.4}$$

$$\textit{Step } 3. \qquad p_i^{k+1} = \pi p_i^k \ . \tag{3.5}$$

Here, $\pi < 1$ is a penalty updating factor. The meaning of the "$\approx$" sign in Step 1 is that the unconstrained minimization problem is only solved approximately, until $\|\nabla_x \mathcal{L}(x,\mu;p)\| \le \varepsilon$, where ε is some prescribed tolerance.

For more details on the PBM method, analysis, and numerical performance, see the references above.

In Step 1 we need to solve, approximately, an unconstrained optimization problem. For this, we will use the Newton method. Therefore, we will need formulas for the gradient and Hessian of $\mathcal{L}$ with respect to the primal variable x:

$$\nabla_x \mathcal{L}(x,\mu;p) = \nabla_x \mathfrak{f}(x) + \sum_{i=1}^{m} \mu_i \varphi'\left(\frac{\mathfrak{g}_i(x)}{p_i}\right) \nabla_x \mathfrak{g}_i(x) \tag{3.6}$$

and

$$\begin{aligned} \nabla^2_{xx} \mathcal{L}(x,\mu;p) =& \nabla^2_{xx} \mathfrak{f}(x) + \sum_{i=1}^{m} \frac{\mu_i}{p_i} \varphi''\left(\frac{\mathfrak{g}_i(x)}{p_i}\right) \nabla_x \mathfrak{g}_i(x) (\nabla_x \mathfrak{g}_i(x))^\top \\ &+ \sum_{i=1}^{m} \mu_i \varphi'\left(\frac{\mathfrak{g}_i(x)}{p_i}\right) \nabla^2_{xx} \mathfrak{g}_i(x). \end{aligned} \tag{3.7}$$

Note that, due to the convexity of the penalized problem (3.1), the Hessian of $\mathcal{L}$ is positive semidefinite for any arguments $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^m_+$.

Ben-Tal and Zibulevsky [7] analyzed one particular choice of the penalty function φ defined as follows:

$$\varphi_{\hat\tau}(\tau) = \begin{cases} \tau + \frac{1}{2}\tau^2, & \tau \ge \hat\tau, \\ -(1+\hat\tau)^2 \log\left(\frac{1+2\hat\tau-\tau}{1+\hat\tau}\right) + \hat\tau + \frac{1}{2}\hat\tau^2, & \tau < \hat\tau \ . \end{cases} \tag{3.8}$$

By setting $\hat\tau = -\frac{1}{2}$, we get a pure (not shifted) logarithmic branch. As this function combines properties of the quadratic penalty function and the logarithmic barrier function, it is called a penalty-barrier function and the resulting algorithm a penalty-barrier multiplier method. This method proved to be very efficient, and we will use it to solve the dual VTS problem.

3.2. PBM for the dual VTS problem. Let us now apply the PBM method to the dual problem (2.1). The augmented Lagrangian for this problem is defined as

$$\begin{aligned}\mathcal{L}(u,\alpha,\underline{\nu},\bar{\nu},\rho,\underline{\mu},\bar{\mu}) = \alpha V - f^\top u - \underline{\rho}^\top \underline{\nu} + \bar{\rho}^\top \bar{\nu} \\ + \sum_{i=1}^m \rho_i p_i \varphi\left(\frac{1}{p_i}\left(\frac{1}{2}u^\top K_i u - \alpha + \underline{\nu}_i - \bar{\nu}_i\right)\right) \\ + \sum_{i=1}^m \underline{\mu}_i \underline{q}_i \varphi\left(\frac{-\underline{\nu}_i}{\underline{q}_i}\right) + \sum_{i=1}^m \bar{\mu}_i \bar{q}_i \varphi\left(\frac{-\bar{\nu}_i}{\bar{q}_i}\right)\end{aligned} \tag{3.9}$$

with Lagrangian multipliers $\rho \in \mathbb{R}^m$, $\underline{\mu} \in \mathbb{R}^m$, and $\bar{\mu} \in \mathbb{R}^m$ and penalty parameters $p \in \mathbb{R}^m$, $\underline{q} \in \mathbb{R}^m$, and $\bar{q} \in \mathbb{R}^m$.

To simplify the notation, let us define the aggregate variable

$$\xi := (u, \alpha, \underline{\nu}, \bar{\nu})$$

and vectors of penalized constraints as

$$\begin{aligned}\tilde{g}_i(\xi) = \tilde{g}_i(u,\alpha,\underline{\nu},\bar{\nu}) &:= \varphi\left(\frac{1}{p_i}\left(\frac{1}{2}u^\top K_i u - \alpha + \underline{\nu}_i - \bar{\nu}_i\right)\right), \quad i = 1,\dots,m, \\ \underline{h}_i(\xi) = \underline{h}_i(u,\alpha,\underline{\nu},\bar{\nu}) &:= \varphi\left(\frac{-\underline{\nu}_i}{\underline{q}_i}\right), \quad i = 1,\dots,m, \\ \bar{h}_i(\xi) = \bar{h}_i(u,\alpha,\underline{\nu},\bar{\nu}) &:= \varphi\left(\frac{-\bar{\nu}_i}{\bar{q}_i}\right), \quad i = 1,\dots,m.\end{aligned}$$

Let $s_i(\xi)$ denote the argument of $\varphi(\cdot)$ in the definition of $\tilde{g}_i(\xi)$ above. In the following, the notation $\tilde{g}_i'(\xi)$ will be understood as $\varphi'(s_i(\xi))$, rather than a composite derivative of $\varphi(s_i(\xi))$ with respect to ξ. We define $\underline{h}_i'(\cdot)$ and $\bar{h}_i'(\cdot)$ analogously, as well as $\tilde{g}''(\cdot)$, $\underline{h}''(\cdot)$, and $\bar{h}''(\cdot)$.

According to (3.6), the gradient of the augmented Lagrangian with respect to the aggregate variable ξ is

$$\nabla_\xi \mathcal{L}(\cdot) = \begin{bmatrix}\mathrm{res}_1 \\ \mathrm{res}_2 \\ \mathrm{res}_3 \\ \mathrm{res}_4\end{bmatrix} = \begin{bmatrix}-f \\ V \\ \underline{\rho} \\ \bar{\rho}\end{bmatrix} + \sum_{i=1}^m \rho_i \tilde{g}_i'(\xi) \begin{bmatrix}K_i u \\ -1 \\ e_i \\ -e_i\end{bmatrix} + \sum_{i=1}^m \underline{\mu}_i \underline{h}_i'(\xi) \begin{bmatrix}0 \\ 0 \\ -e_i \\ 0\end{bmatrix} + \sum_{i=1}^m \bar{\mu}_i \bar{h}_i'(\xi) \begin{bmatrix}0 \\ 0 \\ 0 \\ -e_i\end{bmatrix}. \tag{3.10}$$

To further simplify the notation, we define

$$\begin{aligned}\rho_i' = \rho_i'(\xi) &:= \rho_i \tilde{g}_i'(\xi), & \underline{\mu}_i' = \underline{\mu}_i'(\xi) &:= \underline{\mu}_i \underline{h}_i'(\xi), & \bar{\mu}_i' = \bar{\mu}_i' &:= \bar{\mu}_i \bar{h}_i'(\xi), \\ \rho_i'' = \rho_i''(\xi) &:= \frac{\rho_i}{p_i} \tilde{g}_i''(\xi), & \underline{\mu}_i'' = \underline{\mu}_i'' &:= \frac{\underline{\mu}_i}{\underline{q}_i} \underline{h}_i''(\xi), & \bar{\mu}_i'' = \bar{\mu}_i'' &:= \frac{\bar{\mu}_i}{\bar{q}_i} \bar{h}_i''(\xi).\end{aligned}$$

By (3.7), the Hessian of the augmented Lagrangian will take the form

$$\nabla^2_{(u,\alpha,\underline{\nu},\bar{\nu})^2} \mathcal{L}(\cdot) = \begin{bmatrix}H_{11} & H_{12} & H_{13} & H_{14} \\ H_{12}^\top & H_{22} & H_{23} & H_{24} \\ H_{13}^\top & H_{23}^\top & H_{33} & H_{34} \\ H_{14}^\top & H_{24}^\top & H_{34}^\top & H_{44}\end{bmatrix}, \tag{3.11}$$

where

$$
\begin{aligned}
H_{11} &= \sum_{i=1}^{m} \rho_i'' K_i u u^\top K_i^\top + \sum_{i=1}^{m} \rho_i' K_i, \quad H_{11} \in \mathbb{R}^{n\times n},\\
H_{12} &= -\sum_{i=1}^{m} \rho_i'' K_i u, \quad H_{12} \in \mathbb{R}^{n\times 1},\\
H_{13} &= \left[\rho_1'' K_1 u, \dots, \rho_m'' K_m u\right], \quad H_{13} \in \mathbb{R}^{n\times m},\\
H_{14} &= \left[-\rho_1'' K_1 u, \dots, -\rho_m'' K_m u\right], \quad H_{14} \in \mathbb{R}^{n\times m},\\
H_{22} &= \sum_{i=1}^{m} \rho_i'', \quad H_{22} \in \mathbb{R},\\
H_{23} &= \left[-\rho_1'', \dots, -\rho_m''\right], \quad H_{23} \in \mathbb{R}^{1\times m},\\
H_{24} &= \left[\rho_1'', \dots, \rho_m''\right], \quad H_{24} \in \mathbb{R}^{1\times m},\\
H_{33} &= \operatorname{diag}(\rho_1'' + \underline{\mu}_1'', \dots, \rho_m'' + \underline{\mu}_m''), \quad H_{33} \in \mathbb{R}^{m\times m},\\
H_{34} &= \operatorname{diag}(-\rho_1'', \dots, -\rho_m''), \quad H_{34} \in \mathbb{R}^{m\times m},\\
H_{44} &= \operatorname{diag}(\rho_1'' + \bar{\mu}_1'', \dots, \rho_m'' + \bar{\mu}_m''), \quad H_{44} \in \mathbb{R}^{m\times m}.
\end{aligned}
$$

By (3.4), the Lagrange multipliers in the PBM algorithm are never equal to zero. Hence, the matrices H_{33}, H_{34}, H_{44} are diagonal and positive or negative definite, so we can easily calculate the following inverse of the lower right block of the Lagrangian, which is in turn a block diagonal matrix:

$$
\begin{bmatrix} H_{33} & H_{34} \\ H_{34}^\top & H_{44} \end{bmatrix}^{-1} = \begin{bmatrix} H_{33}^{-1} + H_{33}^{-1} H_{34} Z H_{34}^\top H_{33}^{-1} & -H_{33}^{-1} H_{34} Z \\ -Z H_{34}^\top H_{33}^{-1} & Z \end{bmatrix}
$$

with $Z = (H_{44} - H_{34}^\top H_{33}^{-1} H_{34})^{-1}$. We will require this inverse further below.

Observe that the matrix H_{11} has the same sparsity structure as the "unscaled" stiffness matrix $\sum_{i=1}^m K_i$. Indeed, the only nonzero components of the vector $K_i u$ are those corresponding to indices of nonzero elements of K_i; hence K_i has the same sparsity structure as $(K_i u)(K_i u)^\top$. For this reason, the matrices $H_{13}H_{13}^\top$ and $H_{14}H_{14}^\top$ have the same sparsity structure as H_{11}, and thus $\sum_{i=1}^m K_i$. This property extends to any matrices $H_{13}DH_{13}^\top$ and $H_{14}DH_{14}^\top$, where D is a diagonal matrix.

We now calculate the following Schur complement matrix:

$$
S = \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^\top & H_{22} \end{bmatrix} - \begin{bmatrix} H_{13} & H_{14} \\ H_{23} & H_{24} \end{bmatrix} \begin{bmatrix} H_{33} & H_{34} \\ H_{34}^\top & H_{44} \end{bmatrix}^{-1} \begin{bmatrix} H_{13}^\top & H_{23}^\top \\ H_{14}^\top & H_{24}^\top \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{3.12}
$$

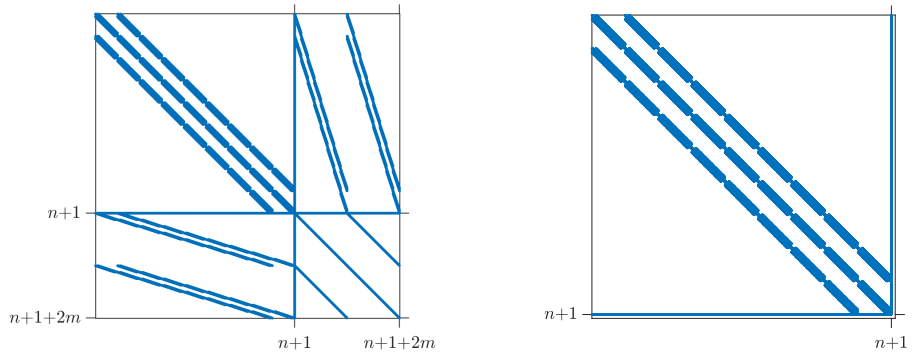
By the previous considerations, the principal $n \times n$ submatrix of S has the same sparsity structure as the stiffness matrix $\sum_{i=1}^m K_i$; the last row and column of S are full. Figure 3.1 shows typical examples of the sparsity structure of the Hessian of the augmented Lagrangian $\nabla^2_{\xi\xi} \mathcal{L}(\cdot)$ in (3.11) and the Schur complement matrix S.

The first step of the PBM algorithm is to solve approximately the unconstrained minimization problem

$$
\min_{u,\alpha,\underline{\nu},\bar{\nu}} \mathcal{L}(u, \alpha, \underline{\nu}, \bar{\nu}, \rho, \underline{\mu}, \bar{\mu})
$$

by the Newton method. In every step of the Newton method, we have to solve the system of linear equations

$$
\nabla^2_{(u,\alpha,\underline{\nu},\bar{\nu})^2} \mathcal{L}(u, \alpha, \underline{\nu}, \bar{\nu}, \rho, \underline{\mu}, \bar{\mu}) \cdot (\Delta u, \Delta\alpha, \Delta\nu) = -\nabla_{(u,\alpha,\underline{\nu},\bar{\nu})} \mathcal{L}(u, \alpha, \underline{\nu}, \bar{\nu}, \rho, \underline{\mu}, \bar{\mu}),
$$



(a) $\nabla^2\mathcal{L}$ (b) S

FIG. 3.1. *Typical sparsity structure of the Hessian of the augmented Lagrangian for the dual topology optimization problem (left) and its Schur complement (right).*

where $(\Delta u, \Delta\alpha, \Delta\nu)$ is the Newton increment and $\Delta\nu := (\Delta\underline{\nu}, \Delta\bar{\nu})$. Equivalently, according to the above development, we can instead solve the reduced system

$$S \begin{bmatrix} \Delta u \\ \Delta\alpha \end{bmatrix} = rhs, \tag{3.13}$$

where, by (3.10),

$$rhs = -\begin{bmatrix} -f \\ V \end{bmatrix} - \sum_{i=1}^{m} \rho_i' \begin{bmatrix} K_i u \\ -1 \end{bmatrix} + \begin{bmatrix} H_{13} & H_{14} \\ H_{23} & H_{24} \end{bmatrix} \begin{bmatrix} H_{33} & H_{34} \\ H_{34}^\top & H_{44} \end{bmatrix}^{-1} \left(\begin{bmatrix} \underline{\rho} \\ \bar{\rho} \end{bmatrix} + \sum_{i=1}^{m} \rho_i' \begin{bmatrix} e_i \\ -e_i \end{bmatrix} + \sum_{i=1}^{m} \underline{\mu}_i' \begin{bmatrix} e_i \\ 0 \end{bmatrix} + \sum_{i=1}^{m} \bar{\mu}_i' \begin{bmatrix} 0 \\ -e_i \end{bmatrix} \right).$$

Recall that the dual problem (2.1) is convex; hence the Hessian of $\mathcal{L}$ is positive semidefinite and, consequently, so is the Schur complement S.

The remaining component $\Delta\nu$ can be reconstructed from the solution to (3.13) as follows:

$$\Delta\nu = -\begin{bmatrix} H_{33} & H_{34} \\ H_{34}^\top & H_{44} \end{bmatrix}^{-1} \left(\begin{bmatrix} \underline{\rho} \\ \bar{\rho} \end{bmatrix} + \sum_{i=1}^{m} \rho_i' \begin{bmatrix} e_i \\ -e_i \end{bmatrix} + \sum_{i=1}^{m} \underline{\mu}_i' \begin{bmatrix} e_i \\ 0 \end{bmatrix} + \sum_{i=1}^{m} \bar{\mu}_i' \begin{bmatrix} 0 \\ -e_i \end{bmatrix} + \begin{bmatrix} H_{13}^\top \\ H_{14}^\top \end{bmatrix} \Delta u + \begin{bmatrix} H_{23}^\top \\ H_{24}^\top \end{bmatrix} \Delta\alpha \right). \tag{3.14}$$

After the augmented Lagrangian has been minimized, we check for convergence. For this, we use the duality gap $\delta(u, \alpha)$ in (2.6), scaled by the dual objective function, henceforth denoted by $d(u, \alpha, \underline{\nu}, \bar{\nu})$, as a measure of optimality. If convergence has not yet been achieved, the multipliers are updated, imposing the safeguard rule used in [7], followed by the penalty parameters.

The PBM method is summarized in Algorithm 3.1. It employs the Newton method with backtracking line search using the Armijo rule; see Algorithm 3.2. The stopping criterion for the Newton method uses the weighted residual term

$$\widetilde{\text{res}}_{PBM} = \frac{\|\text{res}_1\|_2}{\|f\|_2} + \frac{|\text{res}_2|}{V} + \frac{\|\text{res}_3\|_2}{\|\underline{\rho}\|_2 + \|\bar{\rho}\|_2} \tag{3.15}$$

Algorithm 3.1 PBM

Let $0 < \beta < 1$, $0 < \gamma < 1$, $p_{\min}, \underline{q}_{\min}, \bar{q}_{\min} > 0$, $\varepsilon_{\text{PBM}} > 0$, $\varepsilon_{\text{NWT}} > 0$ and $\varepsilon_{\text{NWT}}^{\min} > 0$ be given. Choose initial vectors $(u, \alpha, \underline{\nu}, \bar{\nu})$ and $(\rho, \underline{\mu}, \bar{\mu})$. Set $p = \underline{q} = \bar{q} = e \in \mathbb{R}^m$.

1: **repeat**

2: Minimize the augmented Lagrangian (3.9) with respect to $(u, \alpha, \underline{\nu}, \bar{\nu})$ by Algorithm 3.2 with stopping tolerance ε_{NWT}

3: Compute the duality gap $\delta(u, \alpha)$ by (2.6) and the dual objective function value $d(u, \alpha, \underline{\nu}, \bar{\nu})$

4: **if** $|\delta(u, \alpha)/d(u, \alpha, \underline{\nu}, \bar{\nu})| < \varepsilon_{\text{PBM}}$ **then**

5: STOP

6: **end if**

7: Update the multipliers

$$\rho_i^+ = \rho_i \varphi' \left(\frac{1}{p_i} \left(\frac{1}{2} u^\top K_i u - \alpha + \underline{\nu}_i - \bar{\nu}_i \right) \right), \quad i = 1, \ldots, m,$$

$$\underline{\mu}_i^+ = \underline{\mu}_i \varphi' \left(\frac{\underline{\nu}_i}{\underline{q}_i} \right), \quad \bar{\mu}_i^+ = \bar{\mu}_i \varphi' \left(\frac{-\bar{\nu}_i}{\bar{q}_i} \right), \quad i = 1, \ldots, m$$

8: If necessary, correct the multipliers such that

$$\beta \rho_i \le \rho_i^+ \le \frac{1}{\beta} \rho_i, \quad \beta \underline{\mu}_i \le \underline{\mu}_i^+ \le \frac{1}{\beta} \underline{\mu}_i, \quad \beta \bar{\mu}_i \le \bar{\mu}_i^+ \le \frac{1}{\beta} \bar{\mu}_i, \quad i = 1, \ldots, m$$

and set $\rho = \rho^+$, $\underline{\mu} = \underline{\mu}^+$, $\bar{\mu} = \bar{\mu}^+$.

9: Update the penalty parameters

$$p_i = \max\{\gamma p_i, p_{\min}\}, \quad \underline{q}_i = \max\{\gamma \underline{q}_i, \underline{q}_{\min}\}, \quad \bar{q}_i = \max\{\gamma \bar{q}_i, \bar{q}_{\min}\}$$

for $i = 1, \ldots, m$

10: Update the stopping tolerance for Algorithm 3.2

$$\varepsilon_{\text{NWT}} = \max \left\{ \min \left\{ 100 \cdot \left| \frac{\delta(u, \alpha)}{d(u, \alpha, \underline{\nu}, \bar{\nu})} \right|, \varepsilon_{\text{NWT}} \right\}, \varepsilon_{\text{NWT}}^{\min} \right\}$$

11: **until** convergence

12: Set $\varepsilon_{\text{NWT}} = 10 \cdot \varepsilon_{\text{PBM}}$ and repeat line 2

13: Update ρ as done in line 8

as a measure of feasibility. The stopping parameter is adjusted adaptively in each PBM iteration, and this warrants some clarification. Setting the Newton method tolerance too low in early stages of the PBM method leads to an increase in Newton iterations and thus in computational time without significantly changing the convergence behavior of the PBM. A "soft" tolerance of 100 times the current optimality measure has proven to be a good choice. At the same time, however, we want to guarantee that the final solution has a certain degree of feasibility, which requires the system (3.13) to be solved to a certain accuracy. For this reason, after the final PBM iteration, we run the Newton method one more time with decreased tolerance and then update ρ. For the sake of completeness, it should be noted that the solution $(u, \alpha, \underline{\nu}, \bar{\nu})$ obtained by this additional call to the Newton method is not guaranteed to still satisfy the stopping criterion on line 4 of Algorithm 3.1. It is possible that it

Algorithm 3.2 PBM Newton

Let vectors (u, α, ν) and $(\rho, \underline{\mu}, \bar{\mu})$ be given, using $\nu = (\underline{\nu}, \bar{\nu})$. Let ε_{NWT} be given.

repeat
  Compute matrix S from (3.12) and the corresponding right-hand side
  Solve (approximately) the linear system (3.13) to find $\Delta u, \Delta\alpha$
  Compute $\Delta\nu$ from (3.14) with data $\Delta u, \Delta\alpha$
  Perform backtracking line search with Armijo rule to find step length κ
  Update u, α, ν: $u = u + \kappa\Delta u, \quad \alpha = \alpha + \kappa\Delta\alpha, \quad \nu = \nu + \kappa\Delta\nu$
  if $\widetilde{\text{res}}_{PBM} < \varepsilon_{\text{NWT}}$ **then**
    STOP
  end if
until convergence

was previously only satisfied due to the inaccuracy of the solution.[1] One would then have to run more iterations of the full algorithm to be sure to satisfy the stopping criterion. In the numerical experiments presented in this paper, however, this was not an issue, and $|\delta(u,\alpha)/d(u,\alpha,\underline{\nu},\bar{\nu})|$ remained below the stopping parameter ε_{PBM}. Therefore, we did not include the technicalities of handling this case in Algorithm 3.1.

Our choice of parameters in Algorithm 3.1 was $\beta = \gamma = 0.3$, $p_{\min} = \underline{q}_{\min} = \bar{q}_{\min} = 10^{-8}$, $\varepsilon_{\text{NWT}} = 1$, and $\varepsilon_{\text{NWT}}^{\min} = 10^{-3}$. The initial values were $u = 0$, $\alpha = 1$, $\underline{\nu} = \bar{\nu} = e$, $\rho_i = V/m$ for all $i = 1, \ldots, m$, and $\underline{\mu} = \bar{\mu} = e$.

Note that Algorithm 3.2 is an inexact Newton method, which uses a preconditioned Krylov subspace method, as described later in section 6. Let us reiterate that the principal $n \times n$ submatrix of S in (3.13) has the same sparsity structure as the stiffness matrix $K(\rho)$. This will allow us in section 6 to develop a multigrid preconditioner using the standard prolongation/restriction operators for the stiffness matrix.

4. An IP method for topology optimization. In this section, we describe the primal-dual IP method used to solve (1.1). This involves deriving the linear system to be solved in each iteration and taking Schur complements of this system in order to obtain a system that, firstly, is symmetric positive definite and, secondly, displays a structure that allows a straightforward application of the multigrid method as a preconditioner. In this, we follow [15]. Many features of the algorithm proposed in that reference had to be changed to make it more performant and viable for 3D problems. Therefore, we include all details of the algorithm. We do not recapitulate the basics of primal-dual IP methods and instead refer the reader to [27], to name just one standard piece of literature.

Some notation from the previous section will be reused below for variables that serve a similar purpose. However, the primal and dual variables ρ, u, α, $\underline{\nu}$, and $\bar{\nu}$ have the same meaning in both sections. This is worthwhile to note because it means that the results from the PBM method described in the previous section and the IP method described below are directly comparable.

4.1. Primal-dual IP method for the VTS problem. We start by setting up the KKT conditions for the VTS problem (1.1). Note that the problem exhibits a "hidden convexity," i.e., it is not itself a convex problem but is equivalent to a different, convex problem [6]. The strict feasibility, given for (1.1) by design—see section 1—translates to this equivalent problem. Hence, the Slater condition is satisfied and the

[1] Note that $\delta(u,\alpha)$ is only a valid duality gap for *feasible* solutions u and α.

KKT conditions are necessary and sufficient optimality conditions. They are given by the constraint equations in (1.1) and the equations below.

$$\begin{aligned} \frac{1}{2}u^\top K_i u + \alpha + \underline{\nu}_i - \bar{\nu}_i = 0\,, \quad & i = 1,\dots,m, \\ (\rho_i - \underline{\rho}_i)\underline{\nu}_i = 0\,, \quad & i = 1,\dots,m, \\ (\bar{\rho}_i - \rho_i)\bar{\nu}_i = 0\,, \quad & i = 1,\dots,m\,. \end{aligned}$$

Note that in the above, the Lagrange multipliers for the equilibrium equation constraint $K(\rho)u = f$ have already been eliminated, taking advantage of the fact that the minimum compliance problem is self-adjoint. This means that, due to our choice of objective function, the aforementioned multipliers also satisfy the equilibrium equation—with the the right-hand side only differing by a constant factor. Hence, we can directly identify them with u. See, for example, [8] for details.

The complementarity conditions for the lower and upper bound constraints, i.e., the second and third lines in the system above, are now perturbed by replacing 0 by barrier parameters $r > 0$ and $s > 0$, respectively. The resulting system of equations needs to be solved for fixed r, s in each iteration of the IP algorithm. This is done approximately by performing one iteration of the Newton method. We get the following residual function for the Newton method:

$$\begin{aligned} \operatorname{res}(u,\alpha,\rho,\underline{\nu},\bar{\nu}) \;=\; \begin{bmatrix} \operatorname{res}_1 \\ \operatorname{res}_2 \\ \operatorname{res}_3 \\ \operatorname{res}_4 \\ \operatorname{res}_5 \end{bmatrix} \;=\; & \begin{bmatrix} -f \\ -V \\ 0 \\ -r\,e \\ -s\,e \end{bmatrix} + \sum_{i=1}^m \rho_i \begin{bmatrix} K_i u \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \sum_{i=1}^m \frac{1}{2} u^\top K_i u \begin{bmatrix} 0 \\ 0 \\ e_i \\ 0 \\ 0 \end{bmatrix} \\ & + \sum_{i=1}^m \alpha \begin{bmatrix} 0 \\ 0 \\ e_i \\ 0 \\ 0 \end{bmatrix} + \sum_{i=1}^m \underline{\nu}_i \begin{bmatrix} 0 \\ 0 \\ e_i \\ (\rho_i - \underline{\rho}_i)e_i \\ 0 \end{bmatrix} + \sum_{i=1}^m \bar{\nu}_i \begin{bmatrix} 0 \\ 0 \\ e_i \\ (\bar{\rho}_i - \rho_i)e_i \\ 0 \end{bmatrix}. \end{aligned}$$

Next, we obtain the derivative of the residual function as the block matrix

$$\nabla_{(u,\alpha,\rho,\underline{\nu},\bar{\nu})}\operatorname{res}(\cdot) = \begin{bmatrix} K(\rho) & 0 & B(u) & 0 & 0 \\ 0 & 0 & e^\top & 0 & 0 \\ B(u)^\top & e & 0 & I & -I \\ 0 & 0 & \underline{N} & \underline{P} & 0 \\ 0 & 0 & -\overline{N} & 0 & \overline{P} \end{bmatrix}, \tag{4.1}$$

where $I \in \mathbb{R}^{m\times m}$ is the identity matrix and we use the notation

$$\begin{aligned} B(u) &= [K_1 u, \dots, K_m u]\,, \\ \underline{N} &= \operatorname{diag}(\underline{\nu})\,, \qquad \overline{N} = \operatorname{diag}(\bar{\nu})\,, \\ \underline{P} &= \operatorname{diag}(\rho - \underline{\rho})\,, \qquad \overline{P} = \operatorname{diag}(\bar{\rho} - \rho)\,. \end{aligned}$$

The system matrix ∇res in (4.1) is indefinite. Similar to the procedure in section 3, we can reduce the above system to a positive definite one. We do this in two steps. First, we construct the Schur complement of ∇res with respect to its invertible lower

right block $\left[\begin{smallmatrix} P & 0 \\ 0 & \bar{P} \end{smallmatrix}\right]$. We then in turn form the Schur complement of the result with respect to its lower right block; see [15] for details. This leaves us with the matrix

$$S = \begin{bmatrix} K(\rho) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} B(u) \\ e^\top \end{bmatrix} \left(\underline{P}^{-1} \underline{N} + \bar{P}^{-1} \bar{N} \right)^{-1} \begin{bmatrix} B(u)^\top & e \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{4.2}$$

This matrix is positive definite as long as ρ is strictly feasible and $\underline{\nu}, \bar{\nu} > 0$. Recall that $(K_i u)(K_i u)^\top$ has the same sparsity structure as K_i. Hence, the matrix S in (4.2) has the same sparsity structure as that in (3.12) in the previous section.

In each iteration of the IP method, we approximately solve the nonlinear system

$$\mathrm{res}(u, \alpha, \rho, \underline{\nu}, \bar{\nu}) = 0$$

by performing one iteration of Newton's method. Instead of solving the Newton system

$$\nabla_{(u,\alpha,\rho,\underline{\nu},\bar{\nu})} \mathrm{res}(u, \alpha, \rho, \underline{\nu}, \bar{\nu}) \cdot (\Delta u, \Delta\alpha, \Delta\rho, \Delta\underline{\nu}, \Delta\bar{\nu}) = -\mathrm{res}(u, \alpha, \rho, \underline{\nu}, \bar{\nu}),$$

we solve the equivalent system

$$S \begin{bmatrix} \Delta u \\ \Delta\alpha \end{bmatrix} = rhs, \tag{4.3}$$

where, according to the above reduction of the system,

$$\begin{aligned} rhs = &- \begin{bmatrix} -f \\ -V \end{bmatrix} - \sum_{i=1}^{m} \rho_i \begin{bmatrix} K_i u \\ 1 \end{bmatrix} \\ &- \begin{bmatrix} B(u) \\ e^\top \end{bmatrix} \left(\underline{P}^{-1} \underline{N} + \bar{P}^{-1} \bar{N} \right)^{-1} \left(\mathrm{res}_3 + \underline{P}^{-1} \mathrm{res}_4 - \bar{P}^{-1} \mathrm{res}_5 \right). \end{aligned}$$

From the solution of (4.3), we can reconstruct the increment for ρ using the formula

$$\Delta\rho = -\left(\underline{P}^{-1} \underline{N} + \bar{P}^{-1} \bar{N} \right)^{-1} \left(\mathrm{res}_3 + \underline{P}^{-1} \mathrm{res}_4 - \bar{P}^{-1} \mathrm{res}_5 - B(u)^\top \Delta u - \Delta\alpha\, e \right). \tag{4.4}$$

The increments for the Lagrange multipliers $\underline{\nu}$ and $\bar{\nu}$ are computed based on the stable reduction proposed in [12], with a slight adjustment to account for the upper bound constraints not present in that paper. The multipliers are updated by the following formulas, in the following order:

$$\Delta\bar{\nu} = \frac{1}{\bar{\rho} - \underline{\rho}} \left(\underline{P} \left(B(u)^\top \Delta u + \Delta\alpha\, e \right) - (\underline{N} - \bar{N})\rho - (\mathrm{res}_4 + \mathrm{res}_5 - \underline{P}\, \mathrm{res}_3) \right), \tag{4.5}$$

$$\Delta\underline{\nu} = \Delta\bar{\nu} - B(u)^\top \Delta u - \Delta\alpha\, e - \mathrm{res}_3. \tag{4.6}$$

Once the increments have been obtained, we need to determine an appropriate step length. Our algorithm employs a long step strategy [27] in that it restricts the step length mainly to guarantee feasibility of the next iterate. We do not use the same step length for all increments. Rather, $\Delta\rho$ and Δu use the same step length, the step length for $\Delta\alpha$ is always equal to 1, and different step lengths are calculated for both $\Delta\underline{\nu}$ and $\Delta\bar{\nu}$. For details, see Algorithm 4.1. This strategy proved to be the most effective in numerical experiments.

After each IP iteration, the barrier parameters are updated adaptively. For this, we compute the duality measure for the lower and upper bound constraint,

Algorithm 4.1 Primal-dual IP

Let $\varepsilon_{\text{IP}} > 0$ and $0 < \sigma_r, \sigma_s < 1$ be given. Choose initial vectors (u, ρ) and $(\alpha, \underline{\nu}, \bar{\nu})$. Set barrier parameter update values as $r^+ = \sigma_r \cdot \underline{\nu}^\top(\rho - \underline{\rho})/m$ and $s^+ = \sigma_s \cdot \bar{\nu}^\top(\bar{\rho} - \rho)/m$.

1: **repeat**
2: Solve system (4.3) to obtain $(\Delta u, \Delta\alpha)$
3: Reconstruct $(\Delta\rho, \Delta\bar{\nu}, \Delta\underline{\nu})$ using (4.4)–(4.6)
4: Update barrier parameters: $r = r^+$, $s = s^+$
5: Compute the following step lengths:

$$\kappa_u = \kappa_\rho = \min\left\{0.9 \cdot \min_{\Delta\rho_i > 0} \frac{\bar{\rho}_i - \rho_i}{\Delta\rho_i},\ 0.9 \cdot \min_{\Delta\rho_i < 0} \frac{\underline{\rho}_i - \rho_i}{\Delta\rho_i},\ 1\right\},$$

$$\kappa_{\underline{\nu}} = 0.9 \cdot \min_{\Delta\underline{\nu} < 0} \frac{-\underline{\nu}}{\Delta\underline{\nu}}, \quad \kappa_{\bar{\nu}} = 0.9 \cdot \min_{\Delta\bar{\nu} < 0} \frac{-\bar{\nu}}{\Delta\bar{\nu}}, \quad \kappa_\alpha = 1$$

6: Update all variables:

$$u = u + \kappa_u \Delta u\,, \quad \alpha = \alpha + \kappa_\alpha \Delta\alpha\,, \quad \rho = \rho + \kappa_\rho \Delta\rho\,,$$

$$\underline{\nu} = \underline{\nu} + \kappa_{\underline{\nu}} \Delta\underline{\nu}\,, \quad \bar{\nu} = \bar{\nu} + \kappa_{\bar{\nu}} \Delta\bar{\nu}$$

7: Compute the duality gap $\delta(u, \alpha)$ by (2.6), the objective function $\frac{1}{2} f^\top u$ and the feasibility measure $\widetilde{\text{res}}_{IP}$ by (4.7)
8: **if** $\varepsilon_{\text{IP}} > \delta(u, \alpha)/(\frac{1}{2} f^\top u) > -0.1 \cdot \varepsilon_{\text{IP}}$ and $\widetilde{\text{res}}_{IP} < 10 \cdot \varepsilon_{\text{IP}}$ **then**
9: STOP
10: **end if**
11: Determine barrier parameters for shifted barrier parameter update:

$$r^+ = \sigma_r \cdot \frac{\underline{\nu}^\top(\rho - \underline{\rho})}{m}\,, \quad s^+ = \sigma_s \cdot \frac{\bar{\nu}^\top(\bar{\rho} - \rho)}{m}$$

12: **until** convergence

$$\frac{\underline{\nu}^\top(\rho - \underline{\rho})}{m} \quad \text{and} \quad \frac{\bar{\nu}^\top(\bar{\rho} - \rho)}{m}\,,$$

respectively. We then scale these measures by a constant $0 < \sigma_r < 1$ and $0 < \sigma_s < 1$ to update r and s. At this point, one unconventional feature of our algorithm should be highlighted. The new values for r and s are not used to construct the right-hand side term for the next iteration, but rather for the iteration after that. We found that this "iteration shift," peculiar though it might seem, makes the algorithm significantly more efficient. Indeed, without this shift this version of the code is hardly viable and one requires several Newton iterations per IP iteration instead of just one.

Finally, we require a stopping criterion for the algorithm. Just like in Algorithm 3.1, we use the duality gap $\delta(u, \alpha)$ as a measure of optimality, scaled by the current objective function—the *primal* objective function $\frac{1}{2} f^\top u$, in this case. On top of this, we want to ensure that our solution is feasible to within a certain accuracy. Our feasibility measure is the following sum of weighted residuum norms:

$$\widetilde{\text{res}}_{IP} = \frac{\|\text{res}_1\|_2}{\|f\|_2} + \frac{|\text{res}_2|}{V} + \frac{\|\text{res}_3\|_2}{\|\underline{\nu}\|_2 + \|\bar{\nu}\|_2} + \frac{|e^\top \text{res}_4|}{m} + \frac{|e^\top \text{res}_5|}{m}\,. \tag{4.7}$$

Furthermore, the duality gap should be (nearly) positive, as a negative duality gap points to infeasibility.

Algorithm 4.1 sums up our IP method. The parameter values that we used in our experiments are $\varepsilon_{\text{IP}} = 10^{-5}$, $\sigma_r = \sigma_s = 0.2$. For the initial values, we chose $u = 0$, $\alpha = 1$, $\rho_i = V/m$ for all $i = 1, \ldots, m$ and $\underline{\nu} = \bar{\nu} = e$. The barrier parameters start at $r = s = 10^{-2}$.

5. Optimality Condition (OC) method. To get a broader picture, we will compare the PBM and IP algorithms with the established and commonly used Optimality Condition (OC) method. We will therefore briefly introduce the OC algorithm for VTS. For more details, see [8, page 308] and the references therein.

We adapt the algorithm implemented in the popular code `top88.m` [3]; see Algorithm 5.1. We call it damped OC (DOC) method, due to the exponent $q \le 1$ that shortens the "full" OC step. We use the standard value $q = 0.5$.

Following [3], we use the stopping criterion

$$\|\rho^+ - \rho\|_{\text{inf}} \le \varepsilon_{\text{DOC}},$$

where ρ and ρ^+ are the two most recent iterates. While `top88.m` uses $\varepsilon_{\text{DOC}} = 10^{-2}$, we found that this value is too generous in many 3D examples, resulting in an image that is significantly different from an image obtained with $\varepsilon_{\text{DOC}} \le 10^{-3}$; see Figure 7.2 in section 7, where we address the choice of ε_{DOC} in a bit more detail.

Another parameter we changed, as compared to [3], was the value of the stopping criterion for the bisection method τ_α. In section 7, we use $\tau_\alpha = 0.1\,\varepsilon_{\text{DOC}}$ which leads to a more stable behavior of the DOC and only marginal increase of total CPU time.

The reader may ask about the relation of the DOC stopping criterion (using difference of variables in two subsequent iterations) with the more rigorous criterion based on the duality gap, used in the PBM and IP algorithms. Our experiments revealed a somewhat surprising phenomenon: in most of the problems we solved, the behavior of the two stopping measures was almost identical. This experience justifies

Algorithm 5.1 DOC

Let $\rho \in \mathbb{R}^m$ be given such that $\sum_{i=1}^m \rho_i = V$, $\underline{\rho}_i \le \rho_i \le \bar{\rho}_i$, $i = 1, \ldots, m$. Set $\tau_\alpha = 0.1\,\varepsilon_{\text{DOC}}$ and $q \le 1$.

repeat
  $u = (K(\rho))^{-1} f$
  $\bar{\alpha} = 10000$, $\underline{\alpha} = 0$
  while $\dfrac{\bar{\alpha} - \underline{\alpha}}{\bar{\alpha} + \underline{\alpha}} > \tau_\alpha$ **do**
    $\alpha = (\bar{\alpha} + \underline{\alpha})/2$,
    $\rho_i^+ = \min\left\{\max\left\{\rho_i \dfrac{(u^T K_i u)^q}{\alpha}, \underline{\rho}_i\right\}, \bar{\rho}_i\right\}, \quad i = 1, \ldots, m$
    If $\sum_{i=1}^m \rho_i^+ > V$, then set $\underline{\alpha} = \alpha$; else if $\sum_{i=1}^m \rho_i^+ \le V$, then set $\bar{\alpha} = \alpha$
  end while
  if $\|\rho^+ - \rho\|_{\text{inf}} \le \varepsilon_{\text{DOC}}$ **then**
    STOP
  end if
  $\rho = \rho^+$
until convergence

the use of the DOC stopping criterion and, in particular, the relative fairness of our comparisons of DOC with PBM and IP.

6. Multigrid preconditioned Krylov subspace methods. In the previous sections, we have introduced three algorithms for the solution of the VTS problem, all of which have one thing in common: In every iteration, we have to solve a system of linear equations

$$Az = b\,, \tag{6.1}$$

where $b \in \mathbb{R}^n$ and A is a $n \times n$ symmetric positive definite matrix. In the OC method, A is the stiffness matrix $K(\rho)$ of the linear elasticity problem. In algorithms PBM and IP, A corresponds to the Schur complements S from (3.12) and (4.2), respectively. These latter two matrices have the same sparsity structure. In particular, the principal $(n-1)\times(n-1)$ submatrix of S has the same sparsity structure as the stiffness matrix $K(\rho)$; the last row and column of S are full.

In this section, we will recall an iterative method that is known to be very efficient for linear elasticity problems on well structured finite element meshes. Throughout the section, we will use the notation of (6.1).

6.1. Multigrid preconditioned minimal residual (MINRES) method. We use standard V-cycle correction scheme multigrid method with coarse level problems

$$A_k z^{(k)} = b^{(k)}, \quad k = 1, \dots, \ell - 1\,,$$

where

$$A_{k-1} = I_k^{k-1}(A_k) I_{k-1}^k, \quad b^{(k-1)} = I_k^{k-1}(b^{(k)}), \quad k = 2, \dots, \ell\,.$$

Here we assume that there exist $\ell - 1$ linear operators $I_k^{k-1} : \mathbb{R}^{n_k} \to \mathbb{R}^{n_{k-1}}$, $k = 2, \dots, \ell$, with $n := n_\ell > n_{\ell-1} > \cdots > n_2 > n_1$ and let $I_{k-1}^k := (I_k^{k-1})^T$. As a smoother, we use the Gauss–Seidel iterative method. See, e.g., [13] for details.

Although the multigrid method is very efficient, an even more efficient tool for solving (6.1) may be a preconditioned Krylov type method, whereas the preconditioner consists of one V-cycle of the multigrid method.[2] After experimenting with several Krylov methods, we found that the MINRES algorithm [21] is the most robust for our problems in which the system matrix may converge to a positive semidefinite matrix. We use the standard implementation of MINRES from [4] with a zero initial guess. Since it is used to find Newton steps, the previous solution cannot be expected to be a meaningful initial guess.

6.2. Multigrid MINRES for PBM, IP and OC. In all examples in section 7, we use hexahedral finite elements with trilinear basis functions for the displacement variable u and constant basis functions for the variable ρ, as is the standard in topology optimization. We start with a very coarse mesh and use regular refinement of each element into 8 new elements. The prolongation operators I_{k-1}^k for the variable u are based on a standard 27-point interpolation scheme. For more details, see, e.g., [13]. When solving the linear systems (3.13) and (4.3) in PBM and IP, we also need to prolong and restrict the single additional variable λ; here we simply use the identity.

When we use the regular finite element refinement mentioned above, the stiffness matrix $K(\rho)$ will be sparse and, if a reasonably good numbering of the nodes is used, banded. The number of nonzero elements in a row of $K(\rho)$ does not exceed 81.

[2] We found more than one V-cycle to not be as efficient in terms of overall CPU time.

A typical nonzero structure of K is shown in Figure 3.1(b), if we ignore the additional last column and row in that figure.

As usual, the MINRES method is stopped whenever

$$\|r\|_2 \le \varepsilon_{\mathrm{MR}} \|b\|_2 \,, \tag{6.2}$$

where r is the residuum and b the right-hand side of the linear system, respectively. The choice of the stopping parameter $\varepsilon_{\mathrm{MR}}$ varies between the different algorithms.

Multigrid MINRES for OC. The only degree of freedom in the algorithm is the stopping criterion. The required accuracy of these solutions (such that the overall convergence is maintained) is well documented and theoretically supported in the case of the IP method; it is, however, an unknown in the case of the DOC method; see [2] for detailed discussion. Clearly, if the linear systems in the DOC method are solved too inaccurately, the whole method may diverge or just oscillate around a point which is not the solution.

In all our numerical experiments, we used $\varepsilon_{\mathrm{MR}} = 10^{-4}$. In [15], it was observed that, with this stopping criterion, the number of DOC iterations was almost always the same, whether we used an iterative or a direct solver for the linear systems. Our experiments with 3D problems confirmed this observation.

Multigrid MINRES for PBM. The initial stopping parameter $\varepsilon_{\mathrm{MR}}$ scales with the size of the problem, as it can otherwise be too strict for large problems or too imprecise for small problems. We initialize and update it in the following way:

- We start with $\varepsilon_{\mathrm{MR}} = 10^{-4}\sqrt{n}$.
- Let $\widetilde{\mathrm{res}}_{PBM}$ be the sum of the residua computed in the current step of the PBM Newton Algorithm 3.2, and let $\widetilde{\mathrm{res}}^{+}_{PBM}$ be this sum in the following step. If $\widetilde{\mathrm{res}}^{+}_{PBM} > 0.9\,\widetilde{\mathrm{res}}_{PBM}$, we update $\varepsilon_{\mathrm{MR}} := \max\{0.1\,\varepsilon_{\mathrm{MR}}, 10^{-9}\}$. In other words, we increase the accuracy of the stopping parameter whenever we do not achieve a satisfactory improvement in feasibility and optimality with the current $\varepsilon_{\mathrm{MR}}$.

In our numerical tests, the update had to be done only in a few cases, and the smallest value of $\varepsilon_{\mathrm{MR}}$ needed was $\varepsilon_{\mathrm{MR}} = 10^{-3}$.

Multigrid MINRES for IP. In the IP method, we use an adaptive updating scheme for the stopping parameter, based on the complementarity of the current solution:

- We start with $\varepsilon_{\mathrm{MR}} = 10^{-2}$.
- We compute

$$d = \max\left\{ \max_{i=1,\dots,m} |(\rho_i - \underline{\rho}_i)\underline{\nu}_i| \,, \max_{i=1,\dots,m} |(\bar{\rho}_i - \rho_i)\bar{\nu}_i| \right\}$$

  and set $\varepsilon_{\mathrm{MR}} := \max\{100\,d, 10^{-9}\}$ if this new value is lower than the current $\varepsilon_{\mathrm{MR}}$. The low minimum value of 10^{-9} for $\varepsilon_{\mathrm{MR}}$ has proven to be necessary for convergence in our experiments.

7. Numerical experiments. We now present and compare numerical results for the PBM, IP, and DOC methods. In section 7.1, we focus on a rigorous comparison of the performance of the three algorithms, in terms of both CPU time and required calls to the iterative solver. For this, we look at problems where the number of finite elements is in the order of 10^4 to 10^5. As we will see, the PBM method outperforms both the IP and the DOC methods. When we consider problems with over a million finite elements in section 7.2, we only present results for IP and PBM, since DOC with our required accuracy is no longer practicable.

In the formulation of the VTS problem (1.1), we chose the lower bounds $\underline{\rho}$ to be positive. As far as the underlying physical model is concerned, however, $\underline{\rho} = 0$ would make the most sense, with $\rho_i = 0$ corresponding to an element without material. A lower bound larger than zero might distort the, as it were, physically more accurate results. Yet the strict positivity is required for the positive definiteness of $K(\rho)$ and to bound the condition number of the system matrices arising in the different methods. In our experiments, this turned out to be critical for the OC and IP, but not for the PBM method. Therefore, we generally set $\underline{\rho} = 0$ for PBM only.

The code was implemented in MATLAB, outsourcing certain subroutines to C via MEX files. No parallelization was performed in any of our functions. While the MATLAB inbuilt routines are in general parallelizable, on the BlueBEAR HPC system used to produce the large-scale results in section 7.2, it was limited to a single core.

The design domain for each of our example problems is set up in a way that is based on a multigrid structure. It is a cuboid defined by $m_x \times m_y \times m_z$ cubes of equal size corresponding to the coarse level finite element mesh. We refine the coarse mesh regularly $\ell - 1$ times, giving us ℓ mesh levels in total; each cube element is refined into 8 new elements of equal dimensions. Hence level-2 refinement of a $4\times2\times2$ coarse mesh with 16 elements results in a $8 \times 4 \times 4$ mesh with 128 elements, and level-ℓ refinement of the same coarse mesh results in a mesh with $16 \cdot 8^{\ell-1}$ elements.

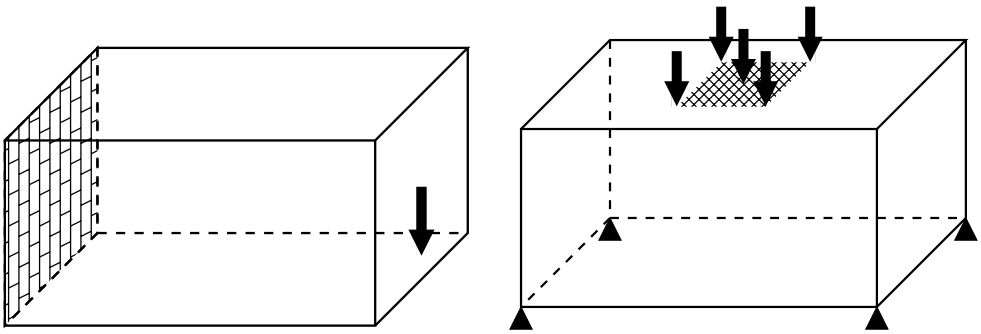
We consider two sets of boundary conditions and loading scenarios, referring to the first one as "cantilever" and to the second one as "bridge"; see Figure 7.1 for the specifications. Cantilever problems have all nodes on the left-hand face fixed in all directions; a load in direction z is applied in the middle of the right-hand face (Figure 7.1(a)). The bridge problems are subject to a uniform load applied on a rectangle centered on the top face; all four corners of the bottom face are fixed in all directions (Figure 7.1(b)).

We adopt the following naming convention for the problems solved in this paper:

CANT-m_x-m_y-m_z-ℓ for a cantilever with a $m_x \times m_y \times m_z$ coarse mesh and ℓ mesh levels;

BRIDGE-m_x-m_y-m_z-ℓ for a bridge with a $m_x \times m_y \times m_z$ coarse mesh and ℓ mesh levels.

7.1. Comparison of PBM, IP, and OC. The problems in this section have been solved on a 2018 MacBook Pro with 2.3GHz dual-core Intel Core i5, Turbo



(a) cantilever (b) bridge

FIG. 7.1. *Boundary conditions and loads for cantilever and bridge problems.*

TABLE 7.1
Example CANT-16-2-2-5 by different methods. Problem dimensions: $m = 262\,144$, $n = 836\,352$.

Method	Stop tol	Iterations Nwt/OC	Iterations MINRES	CPU time [s] Total	CPU time [s] Lin solv	Obj fun
PBM	10^{-5}	42	156	916	317	**66.192**8136
PBM	10^{-6}	48	259	1110	420	**66.19273**18
IP	10^{-5}	36	4992	6510	5670	**66.19272**49
IP($\underline{\rho} = 10^{-3}$)	10^{-5}	29	1243	1510	1070	66.1988863
DOC	10^{-2}	38	394	1160	883	**66.2**107223
DOC	10^{-3}	226	2462	6710	5020	**66.193**4912
DOC	10^{-5}	2759	30325	82500	61900	**66.19272**72

Boost up to 3.6GHz, and 16GB RAM. This allowed us to properly compare the CPU timing, but it also prevented us from solving large-scale problems, due to memory limitations. The results for those problems, run on a HPC computer, are reported in the next section.

Example CANT-16-2-2-5. In Table 7.1 we present results for problem CANT-16-2-2-5 with 262 144 finite elements. The lower bound for ρ was set to zero for the PBM method and to $\underline{\rho} = 10^{-7}$ for the IP and DOC methods.

Each table row shows the results for a certain method and stopping parameter. They are given in terms of the total number of linear systems solved;[3] the total number of MINRES iterations; the total CPU time needed to solve the problem; the CPU time spent on solving the linear systems; and the final value of the primal objective function, where the accurate digits[4] are in bold.

Because the IP method had difficulties getting below our stopping threshold $\varepsilon_{\text{IP}} = 10^{-5}$, we also ran this method with $\underline{\rho} = 10^{-3}$ for comparison, since this improves the conditioning of the system matrix. The resulting objective value is not comparable with the other values and is thus grayed out.

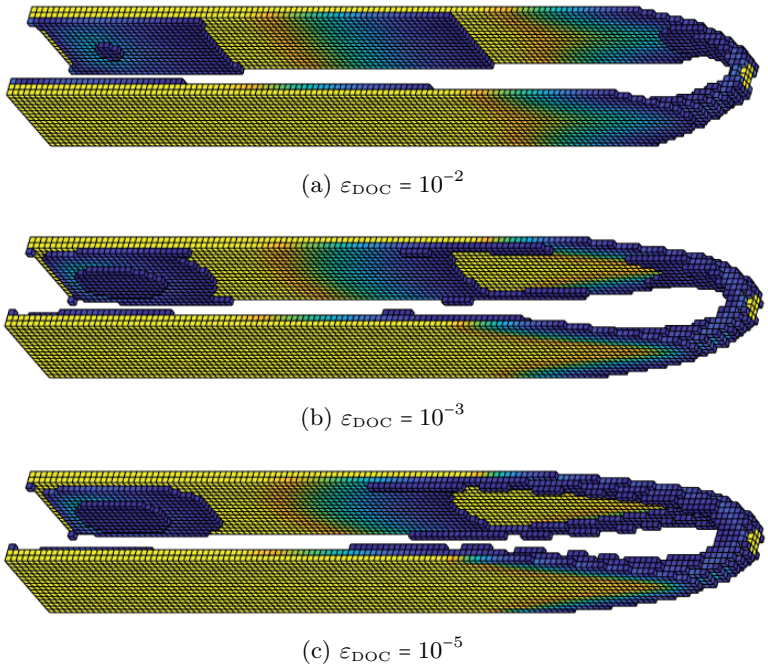
We ran the DOC method with three different stopping tolerances. While $\varepsilon_{\text{DOC}} = 10^{-2}$ would be used to mimic the `top88` code, we can see in Figure 7.2 that the final result delivered with this tolerance is by no means optimal and clearly differs from that obtained with $\varepsilon_{\text{DOC}} = 10^{-5}$ (for better transparency, Figure 7.2 present results of a smaller problem CANT-16-2-2-4). Decreasing ε_{DOC} to 10^{-3} improves the result, but the image is still visibly different from the optimal one. This is despite the five correct significant digits in the objective function, reached by DOC with $\varepsilon_{\text{DOC}} = 10^{-3}$. The results produced by PBM and IP were "visually identical" to that for DOC with $\varepsilon_{\text{DOC}} = 10^{-5}$ in Figure 7.2(c). (Of course, this "visual comparison" is not rigorous, but, in the end, the image is the required result of topology optimization in practice; a rigorous comparison is given in Table 7.1.)

We also ran the PBM method with a lower stopping tolerance $\varepsilon_{\text{PBM}} = 10^{-6}$ to demonstrate that the method can reach higher precision with only relatively few additional iterations.

The numbers in Table 7.1 show that PBM clearly outperforms the other two methods, with respect both to the number of MINRES iterations and to the CPU time required by the whole algorithm and the linear solver only. It is even faster than

[3] This is equal to the number of Newton iterations in the case of the PBM method. For the other two algorithms, there is no difference between the number of "outer" iterations and the number of Newton iterations.

[4] Digits are assumed to be accurate when the different methods all appear to converge to them.



(a) $\varepsilon_{\mathrm{DOC}} = 10^{-2}$

(b) $\varepsilon_{\mathrm{DOC}} = 10^{-3}$

(c) $\varepsilon_{\mathrm{DOC}} = 10^{-5}$

FIG. 7.2. *CANT*-16-2-2-4*, DOC result with* $\varepsilon_{\mathrm{DOC}} = 10^{-2}$*,* $\varepsilon_{\mathrm{DOC}} = 10^{-3}$*,* $\varepsilon_{\mathrm{DOC}} = 10^{-5}$*. Figure* (c) *is identical with IP and PBM results. Only elements with density values of* $\rho_i > 0.1$ *are shown in order to make the differences visible.*

the DOC method with the very relaxed stopping tolerance $\varepsilon_{\mathrm{DOC}} = 10^{-2}$, at the same time delivering a solution of much higher quality.

Below are some further, detailed observations:

- The PBM iterations are very robust in terms of MINRES iterations needed to solve the linear systems. Up to the very last PBM iterations, MINRES only requires 1–3 steps to reach the required accuracy. Even in the last PBM iterations, the number of MINRES steps typically does not exceed 15–20. One reason for this is presumably that the updating scheme for the MINRES tolerance $\varepsilon_{\mathrm{MR}}$ (see section 6) only rarely needs to update the value. With $\varepsilon_{\mathrm{MR}}$ thus decreasing only very slowly, the linear systems never have to be solved to a very high accuracy. Still, the PBM solution displays the required optimality and feasibility.
- The IP method is much more sensitive to ill-conditioning. While in the first IP iterations MINRES only requires 1–2 steps, this number then quickly increases when nearing the required IP stopping criterion. In the CANT-16-2-2-5 problem with $\underline{\rho} = 10^{-3}$, the number of MINRES steps in the IP Newton iterations grew as follows: 1–1–1–1–1–1–1–1–1–1–1–2–3–3–6–5–7–11–13–23–35–49–55–25–66-466–149–314.
- The number of MINRES steps in every DOC iteration is almost constant. In the CANT-16-2-2-4 problem, this number was between 8 and 11 in the first 49 DOC iterations and 12 for all remaining DOC iterations, even with the stopping tolerance $\varepsilon_{\mathrm{DOC}} = 10^{-5}$. The total number of DOC iterations, however, grows dramatically when higher precision in the stopping criterion is required.

TABLE 7.2
Example BRIDGE-4-2-2-6 solved by different methods. Problem dimensions: $m = 524\,288$, $n = 1\,635\,063$.

Method	Stop tol	Iterations Nwt/OC	Iterations MINRES	CPU time [s] Total	CPU time [s] Lin solv	Obj fun
PBM	10^{-5}	57	330	2710	1020	**42.000**2293
PBM	10^{-6}	62	423	3000	1190	**42.0001**561
IP	10^{-5}	49	2919	6040	4320	**42.000**2076
IP($\underline{\rho} = 10^{-3}$)	10^{-5}	51	2965	6210	4440	42.0027639
DOC	10^{-2}	99	1454	6800	5330	**42.00**14281
DOC	10^{-3}	278	4139	19200	15100	**42.000**2175
DOC	10^{-5}	659	9854	45900	36100	**42.0001**523

- Because of the way that ρ is computed in the different algorithms, the volume constraint is not satisfied to the same degree of accuracy in each case. The OC method yields the most accurate ρ with respect to the volume constraint, while the PBM solution generally gives $\sum_i \rho_i > V$. The PBM solution deviation from V was never more than one permille in our experiments.

Example BRIDGE-4-2-2-6. We now present some results of the BRIDGE problem. Table 7.2 shows the iteration numbers and CPU times for BRIDGE-4-2-2-6 with 524 288 finite elements. Compared to CANT-16-2-2-x, the stiffness matrix in these problems (and thus the Schur complement for each method) has a higher condition number due to the different shape of the computational domain.

7.2. Large-scale problems. In this section, we do not include the CPU times needed to solve the example problems. This is because they were solved on the Linux HPC BlueBEAR with 2000 cores of different types, with up to 498 GB RAM per core. We did not have any control over which cores were used for which job, so that the time statistics could not be used for reliable performance comparison. Furthermore, recall that MATLAB only ran on a single core on BlueBEAR, so that the total computation time for any example would most likely not be competitive compared with any parallelized code.

We present results for the PBM and IP algorithms only. As we have seen in section 7.1, they are both several times faster than the DOC method for the same degree of accuracy. This does not improve with larger problem sizes, which means that the OC method might take several days to solve a problem which is solved in just a few hours by the PBM method. To solve the BRIDGE-4-2-2-5 and BRIDGE-4-2-2-6 problems, for example, the OC method requires roughly 17 times as much CPU time as the PBM method. This factor is even larger for CANT-16-2-2-4 and CANT-16-2-2-5. Comparisons for CANT-4-2-2-5 and CANT-4-2-2-5, which are not included here, gave a factor of over 20.

As before, we set $\underline{\rho} = 0$ for the PBM method. For IP, we chose $\underline{\rho} = 10^{-3}$, as ill-conditioning becomes critical in the large-scale problems covered in this section. Even with this lower bound, IP was not able to solve all of the examples we considered. When it failed, no convergence was apparent once the duality gap had gotten below a certain threshold, which was typically still two or three orders of magnitude too large for the stopping criterion.

The same two problems are considered as in the previous section, namely, CANT-m_x-m_y-m_z-ℓ and BRIDGE-m_x-m_y-m_z-ℓ. In this section, we fix the width and height of the design domain to $m_y = m_z = 2$ and vary the length $m_x = 2, 4, 6, 8$. We

TABLE 7.3
Example CANT-m_x-m_y-m_z-ℓ solved by IP and PBM. Overall IP/PBM iterations, Newton iterations, and MINRES iterations. Nondefault parameters: [1] *$\gamma = \beta = 0.5$ and initial $\varepsilon_{\mathrm{MR}} = 10^{-5}\sqrt{n}$.*

Problem dimensions			IP		PBM		
m_x-m_y-m_z-ℓ	m	n	IP/Nwt	MR	PBM	Nwt	MR
2-2-2-5	32 768	104 544	31	368	16	50	175
4-2-2-5	65 536	209 088	28	570	15	57	153
6-2-2-5	98 304	313 632	26	467	14	45	84
8-2-2-5	131 072	418 176	27	489	14	45	115
2-2-2-6	262 144	811 200	46	1195	18	60	141
4-2-2-6	524 288	1 622 400	42	2465	17	59	118
6-2-2-6	786 432	2 433 600	39	1015	17	66	157
8-2-2-6	1 048 576	3 244 800	39	1079	16	57	88
2-2-2-7	2 097 152	6 390 144	71	2383	22	66	70
4-2-2-7	4 194 304	12 780 288	54	3543	20	57	67
6-2-2-7	6 291 456	19 170 432	57	2667	19	60	68
8-2-2-7	8 388 608	25 560 576	58	2335	29 [1]	64 [1]	100 [1]

TABLE 7.4
Example BRIDGE-m_x-m_y-m_z-ℓ solved by IP and PBM. Overall IP/PBM iterations, Newton iterations, and MINRES iterations. When $\widetilde{\mathrm{res}}_{PBM}$ in the final PBM iteration did not go below $\varepsilon_{\mathrm{NWT}}$, the value at the accepted solution is given. Nondefault parameters: [1] *$\gamma = \beta = 0.5$;* [2] *initial $\varepsilon_{\mathrm{MR}} = 10^{-5}\sqrt{n}$;* [3] *initial $\varepsilon_{\mathrm{MR}} = 10^{-5}\sqrt{n}$ and $\varepsilon_{\mathrm{NWT}} = 0.1$.*

Problem dimensions			IP		PBM			
	m	n	IP	MR	PBM	Nwt	MR	$\widetilde{\mathrm{res}}_{PBM}$
2-2-2-5	32 768	107 799	24	387	15	49	220	
4-2-2-5	65 536	212 343	25	590	14	45	155	
6-2-2-5	98 304	316 887	26	778	15	55	309	
8-2-2-5	131 072	421 431	25	1050	13	47	184	
2-2-2-6	262 144	823 863	41	2029	15	56	263	
4-2-2-6	524 288	1 635 063	49	2919	15	57	330	
6-2-2-6	786 432	2 446 263	–	–	15	61	466	
8-2-2-6	1 048 576	3 257 463	–	–	15	62	592	
2-2-2-7	2 097 152	6 440 055	91	4744	26 [1]	109 [1]	1134 [1]	1.14×10^{-4}
4-2-2-7	4 194 304	12 830 199	–	–	26 [1]	99 [1]	718 [1]	2.68×10^{-4}
6-2-2-7	6 291 456	19 220 343	–	–	25 [1,3]	98 [1,3]	743 [1,3]	2.22×10^{-4}
8-2-2-7	8 388 608	25 610 487	–	–	25 [1,2]	97 [1,2]	707 [1,2]	1.17×10^{-3}

ran the code with $\ell = 5, 6, 7$ mesh levels. Tables 7.3 and 7.4 show the results for CANT-m_x-m_y-m_z-ℓ and BRIDGE-m_x-m_y-m_z-ℓ in terms of iteration numbers.

The optimal designs produced by the PBM method can be seen in Figures 7.3 and 7.4. The VTS solution typically has a large "gray area," i.e., ρ_i is well within the interval $[\,\underline{\rho}, \bar{\rho}\,]$ for the majority of elements. This makes it less straightforward to interpret the solution as a discrete design than it is in the case of the SIMP formulation [8]. We must determine a cutoff value ρ^* such that all elements with $\rho_i < \rho^*$ are ignored. As the design domain is elongated, the density distribution further does not change in a linear fashion. Rather, the gray area is spread disproportionately more thin while most solid elements are clustered along the boundary. Therefore, instead of choosing a constant cutoff value, we found that the most consistent way to plot the results was to consider only the densest elements which add up to a fixed proportion cV of allowed volume, where we chose $c = 0.8$.

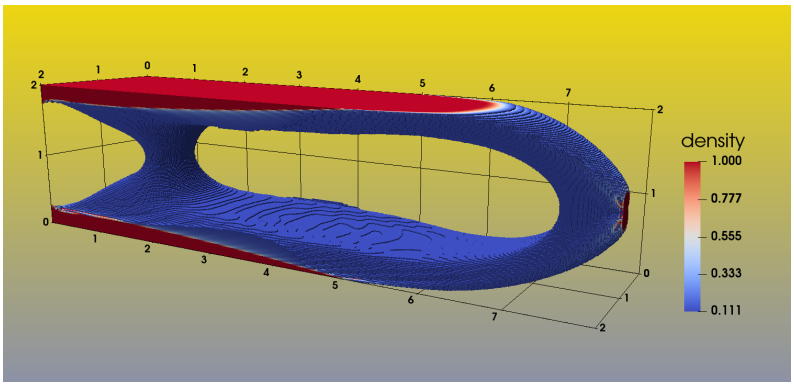


FIG. 7.3. *Optimal density ρ for CANT-8-2-2-7. The elements with the lowest density values are hidden such that the visible element densities add up to $0.8 \cdot V$.*

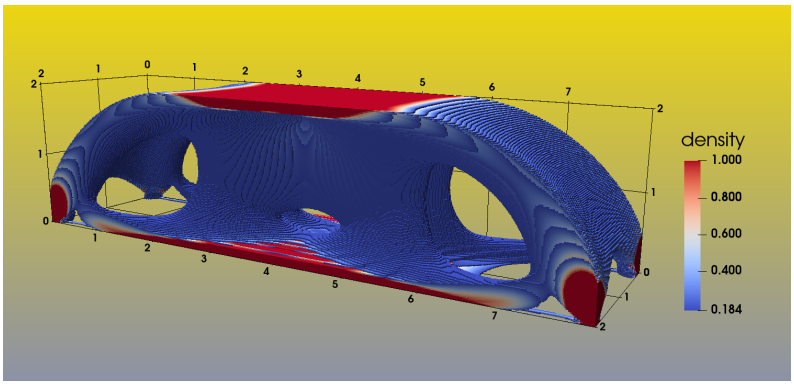


FIG. 7.4. *Optimal density ρ for BRIDGE-8-2-2-7. The elements with the lowest density values are hidden such that the visible element densities add up to $0.8 \cdot V$.*

To solve some of the examples by the PBM method, we had to deviate from the choice of parameters specified earlier. For some examples with $\ell = 7$ refinement levels, we set $\gamma = \beta = 0.5$, rather than $\gamma = \beta = 0.3$. Otherwise, the penalty parameters are scaled down too fast for these largest examples, so that the system becomes too ill-conditioned before we reach optimality. For the specific example CANT-8-2-2-7, we set the initial $\varepsilon_{\text{MR}} = 10^{-5}\sqrt{n}$, because this additional accuracy was required for convergence. Such nondefault parameter choices are indicated in Tables 7.3 and 7.4.

It needs to be said that even with such adjustments, the PBM algorithm did not solve all problems to the specified accuracy. For all BRIDGE-m_x-m_y-m_z-ℓ examples with $\ell = 7$, it failed either close to or in the last iteration, after $\delta(u, \alpha)/\frac{1}{2}f^{\top}u$ had dropped below $\varepsilon_{\text{PBM}} = 10^{-5}$ and ε_{NWT} had been set to 10^{-4}. The residual term $\widetilde{\text{res}}_{PBM}$, defined in (3.15), did not go below ε_{NWT} as required. This was because at a certain point, the approximate solutions of the reduced Newton system (3.13) were no longer directions of descent, presumably due to numerical errors. In these cases, we accepted the, as it were, nearly optimal solutions at which the algorithm stalled. The iteration numbers we list in the table are those after which no further change in residual values is seen. Note that $\widetilde{\text{res}}_{PBM}$ was well below 10^{-3} for all cases except BRIDGE-8-2-2-7, and the scaled duality gap of the accepted solution was always below 10^{-5}.

It is evident from Tables 7.3 and 7.4 that the PBM method is both more efficient and more robust than the IP method. In both cases, the use of a multigrid preconditioner for the MINRES solver achieves the desired result in that the number of MINRES iterations grows sublinearly with the size of the system, if at all. The CANT-m_x-m_y-m_z-ℓ example even displays a decrease in MINRES iterations with larger system size in some cases. However, this is probably not representative, and a possible explanation involves the parameter ε_{MR}: since its initial value scales with the problem size, it might simply be chosen lower than necessary for the smaller problems.

8. Conclusion. In this paper, we proposed a PBM method to solve the dual of the VTS formulation of the minimum compliance topology optimization problem. We compared it with the DOC method, one of the most popular methods for topology optimization, on the one hand, and with the IP method as an established method for general convex problems, on the other. The implementations of both the PBM and IP algorithms were tailored to the specific problem. All three methods used a multigrid preconditioned MINRES solver for the linear systems arising in each iteration.

In our numerical experiments, the PBM method clearly came out on top. It was around 20 times faster in terms of CPU time than the OC method when requiring the same degree of optimality. Even when using a very generous stopping criterion in the OC method—one that yields visibly suboptimal results—PBM was still faster.

The IP method suffers from the characteristic ill-conditioning of the system matrix, which in some of our experiments prevented convergence altogether. Here, PBM proved to be much more robust, in addition to being considerably faster. Still, convergence was not guaranteed for all large-scale examples when sticking to the strictest stopping criterion. Judging by the symmetry and smoothness of the final design, the results were still satisfactory. Overall, the convergence behavior of the PBM method seems to be sensitive to changes in parameters such as stopping tolerances or scaling parameters. A thorough parameter study might further improve the algorithm.

We did not consider the DOC method for such large-scale problems, as its expected computation time simply disqualified it as a competitor. It is however possible that it would eventually converge even for those problems where PBM does not. Note that this would most likely take days or even weeks, as compared to the typical (successful) PBM run which took less than 12 hours. Since the DOC does not feature multipliers or barrier/penalty parameters tending to 0, it is not as susceptible to ill-conditioning as the PBM or IP method. This means that the advantage of DOC, when compared with PBM, could be reliability, albeit at the price of serious inefficiency.

Acknowledgment. The authors would like to thank Michael Stingl for the use of the MATLAB routines used to visualize the results in section 7.1.

REFERENCES

[1] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, *Giga-voxel computational morphogenesis for structural design*, Nature, 550 (2017), pp. 84–86. ISSN 1476-4687.

[2] O. Amir, N. Aage, and B. S. Lazarov, *On multigrid-CG for efficient topology optimization*, Struct. Multidiscip. Optim., 49 (2014), pp. 815–829.

[3] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, *Efficient topology optimization in MATLAB using* 88 *lines of code*, Struct. Multidiscip. Optim., 43 (2011), pp. 1–16.

[4] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994.

[5] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM J. Optim., 3 (1993), pp. 322–358.
[6] A. Ben-Tal and M. Teboulle, *Hidden convexity in some nonconvex quadratically constrained quadratic programming*, Math. Program., 72 (1996), pp. 51–63.
[7] A. Ben-Tal and M. Zibulevsky, *Penalty/barrier multiplier methods for convex programming problems*, SIAM J. Optim., 7 (1997), pp. 347–366.
[8] M. Bendsøe and O. Sigmund, *Topology Optimization. Theory, Methods and Applications*, Springer-Verlag, Heidelberg, 2003.
[9] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comput., 31 (1977), pp. 333–390.
[10] W. Briggs, V. E. Henson, and S. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, PA, 2000.
[11] I. Ekeland and R. Témam, *Convex Analysis and Variational Inequalities*, Classics Appl. Math., SIAM, Philadelphia, PA, 1999.
[12] R. W. Freund and F. Jarre, *A QMR-based interior-point algorithm for solving linear programs*, Math. Program., 76 (1997), pp. 183–210.
[13] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Cham, 1985.
[14] F. Jarre, M. Kočvara, and J. Zowe, *Optimal truss design by interior-point methods*, SIAM J. Optim., 8 (1998), pp. 1084–1107.
[15] M. Kočvara and S. Mohammed, *Primal-dual interior point multigrid method for topology optimization*, SIAM J. Sci. Comput., 38 (2016), pp. B685–B709.
[16] M. Kočvara and M. Stingl, *PENNON: A code for convex nonlinear and semidefinite programming*, Optim. Methods Softw., 18 (2003), pp. 317–333.
[17] M. Kočvara and M. Stingl, *On the solution of large-scale SDP problems by the modified barrier method using iterative solvers*, Math. Program., 109 (2007), pp. 413–444.
[18] M. Kočvara, *Truss topology design by conic linear optimization*, in Advances and Trends in Optimization with Engineering Applications, T. Terlaky, M. F. Anjos, and S. Ahmed, eds., SIAM, Philadelphia, PA, 2017, pp. 135–147.
[19] M. Kočvara, M. Zibulevsky, and J. Zowe, *Mechanical design problems with unilateral contact*, ESAIM Math. Model. Numer. Anal., 32 (1998), pp. 255–281.
[20] B. Maar and V. Schulz, *Interior point multigrid methods for topology optimization*, Struct. Multidiscip. Optim., 19 (2000), pp. 214–224.
[21] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[22] R. Polyak, *Modified barrier functions (theory and methods)*, Math. Program., 54 (1992), pp. 177–222.
[23] R. Polyak and M. Teboulle, *Nonlinear rescaling and proximal-like methods in convex optimization*, Math. Program., 76 (1997), pp. 265–284.
[24] M. Stingl, *On the solution of nonlinear semidefinite programs by augmented Lagrangian methods*, Ph.D. thesis, University of Erlangen, Erlangen, Germany, 2006.
[25] K. Svanberg, *A class of globally convergent optimization methods based on conservative convex separable approximations*, SIAM J. Optim., 12 (2002), pp. 555–573.
[26] M. Teboulle, *Entropic proximal mappings with applications to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.
[27] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.